

---

<b>CURSO:</b>	<b>ENGENHARIA DE SOFTWARE</b>		
<b>DISCIPLINA:</b>	<b>Estruturas de Dados e Algoritmos</b>	<b>TURMA:</b>	<b>A</b>
<b>SEMESTRE:</b>	<b>2013.2</b>	<b>CRÉDITOS:</b>	<b>4</b>
<b>PROFESSOR:</b>	<b>Giovanni Almeida Santos</b>		

---

## ROTEIRO PARA EXECUÇÃO DOS PROGRAMAS

---

### 1. INSTALAÇÃO DO EJUDGE

---

O ejudge deverá ser utilizado para verificar se o seu programa gera os resultados esperados de acordo com os dados fornecidos como entrada. Dessa forma, você tem como saber se o seu programa resolve adequadamente o problema que foi proposto.

Os requisitos para a execução do ejudge são: sistema operacional Linux e ambiente Python.

Para instalar o ejudge, siga os passos abaixo:

- 1) Baixe o arquivo compactado ejudge que está disponível na página da disciplina no moodle;
- 2) Descompacte o arquivo 'ejudge.tar.gz':

```
$ tar vxzpf ejudge.tar.gz
```

- 3) Entre no diretório 'ejudge':

```
$ cd ejudge
```

- 4) Execute o script de instalação:

```
$ ./install
```

Após a instalação, será criada a pasta ejudge dentro do diretório home do usuário (ex: /home/alunos/ejudge).

---

### 2. EXECUÇÃO DOS PROGRAMAS

---

Os nomes dos programas criados por você deverão seguir a seguinte regra:

**<numero\_do\_problema>\_<sua\_matricula>.<extensão>**

Exemplos:

**1\_12000666.c**

**2\_12000666.cpp**

Para cada problema, serão fornecidos três arquivos com dados de entrada e três arquivos com os resultados esperados. Os nomes dos arquivos seguem o padrão abaixo:

**in<numero\_do\_problema>s.txt** → arquivo com número de entradas reduzido (4 entradas)

**in<numero\_do\_problema>e.txt** → arquivo com número de entradas estendido (10 entradas)

**in<numero\_do\_problema>c.txt** → arquivo com número de entradas completo (1000 entradas)

**out<numero\_do\_problema>s.txt** → arquivo com resultados esperados reduzido

**out<numero\_do\_problema>e.txt** → arquivo com resultados esperados estendido

**out<numero\_do\_problema>c.txt** → arquivo com resultados esperados completo

Exemplos:

**in2s.txt, in2e.txt, in2c.txt**

**out2s.txt, out2e.txt, out2c.txt**

Estes arquivos deverão ser copiados para a pasta ejudge que foi criada do diretório home do usuário (ex: /home/alunos/ejudge).

Após implementar o seu programa, você poderá testá-lo com o ejudge, executando o seguinte comando:

**\$ ejudge <numero\_do\_problema>\_<sua\_matricula>.<extensão>**

Ao executar o comando acima, o ejudge irá compilar o seu programa, executá-lo com os valores de entrada fornecidos e comparar os resultados gerados com as saídas esperadas.

Um programa correto poderá gerar as seguintes saídas:

```
Verificando o nome do arquivo: Ok!
Compilando o arquivo: Ok!
Verificando os inputs principais: Ok!
Verificando os inputs complementares: Ok!
Corrigindo a questao: Arquivos para correcao nao disponiveis!
Deseja enviar o arquivo? (s/n): n
[Erro]: Arquivo nao enviado!
```

Figura 1 - Saída quando somente há os arquivos reduzido e estendido

```
Verificando o nome do arquivo: Ok!
Compilando o arquivo: Ok!
Verificando os inputs principais: Ok!
Verificando os inputs complementares: Ok!
Corrigindo a questao: Ok!
Deseja enviar o arquivo? (s/n): n
[Erro]: Arquivo nao enviado!
```

Figura 2 - Saída quando estão presentes todos os três arquivos

---

### 3. OBSERVAÇÕES

---

O ejudge utiliza o padrão ANSI da linguagem C. Dessa forma, algumas regras vão ter que ser seguidas durante a escrita dos programas. Por exemplo:

- 1) a função main deve ter o tipo de retorno int;
- 2) os comentários no código deverão ser escritos com `/* */`, mesmo aqueles em apenas uma linha.

Os comandos utilizados pelo ejudge para compilação são os seguintes:

```
$ gcc -g -o prog -W -Wall -pedantic -ansi -I. arquivo.c -lm
```

```
$ g++ -g -o prog -W -Wall -pedantic -ansi -I. arquivo.cpp -lm
```